IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

In re Application of: Liu, et al.	§	
	§	
	§	Group Art Unit: 2441
Serial No.: 10/726,002	§	
	§	
Confirmation No.: 5240	§	
	§	Examiner: Brian J. Gillis
Filed: December 1, 2003	§	
	§	
For: METHOD AND APPARATUS FOR VIRTUALIZING NETWORK RESOURCES	§	

MAIL STOP RCE
Commissioner for Patents
P.O. Box 1450
Alexandria, VA 22313-1450

DECLARATION UNDER 37 C.F.R. § 1.131

Dear Sir:

We, Zhen Liu, George Popescu, and Sambit Sahu, hereby declare as follows:

1. We are the inventors of the above-captioned patent application.

2. The invention of pending claims 1 and 12-23 was conceived of prior to May 27, 2003, and filed with due diligence from prior to May 27, 2003, to filing of the present application on December 1, 2003.

3. Enclosed herewith in support of this declaration is an invention disclosure (Exhibit A) dated prior to May 27, 2003. Some information has been redacted.

We further declare that all statements made herein of our own knowledge are true and that all statements made on information and belief are believed to be true; and further, that these statements were made with the knowledge that willful false

statements and the like so made are punishable by fine or imprisonment, or both, under Section 1001 of Title 18 of the United States Code and that such willful false statements may jeopardize the validity of the application or any patent issuing thereon.

Zhen Liu	Date
George Popescu	Date
Sambit Sahu	7/7/09 Date

EXHIBIT A

Disclosure YOR8-2003-0292

Prepared for and/or by an IBM Attorney - IBM Confidential

Created By Sambit Sahu On [redacted]
Last Modified By Jeanne M Jordan On [redacted]

Required fields are marked with the asterisk (*) and must be filled in to complete the form .

*Title of disclosure (in English)
Virtualization of Network Resources for Distributed Interactive Applications

Summary

Status	Final Decision (File)
Final Deadline	
Final Deadline Reason	
Docket Family	YOR9-2003-0522

[redacted]

Inventors with a Blue Pages entry

Inventors: Sambit Sahu/Watson/IBM, Zhen Liu/Watson/IBM, George Popescu/Watson/IBM

Inventor Name	Inventor Serial	Div/Dept	Inventor Phone	Manager Name
[redacted]				

> denotes primary contact

Inventors without a Blue Pages entry

IDT Selection

Attorney/Patent Professional IDT Team [redacted]

Response Due to IP&L [redacted]

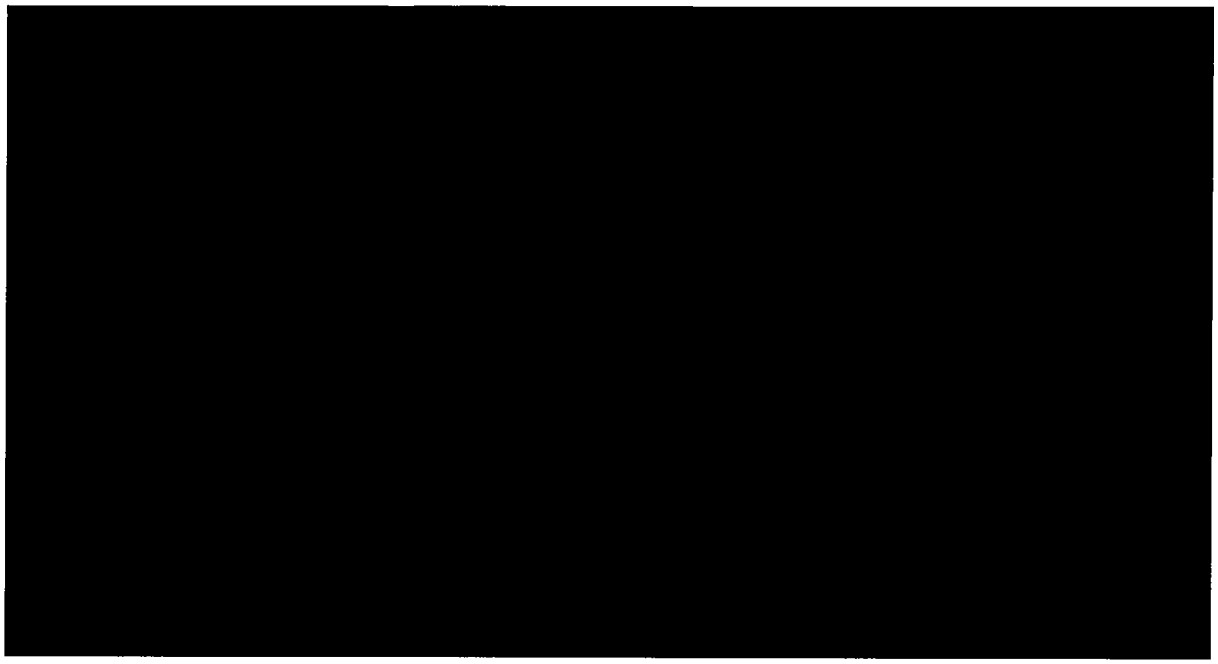
*Main Idea

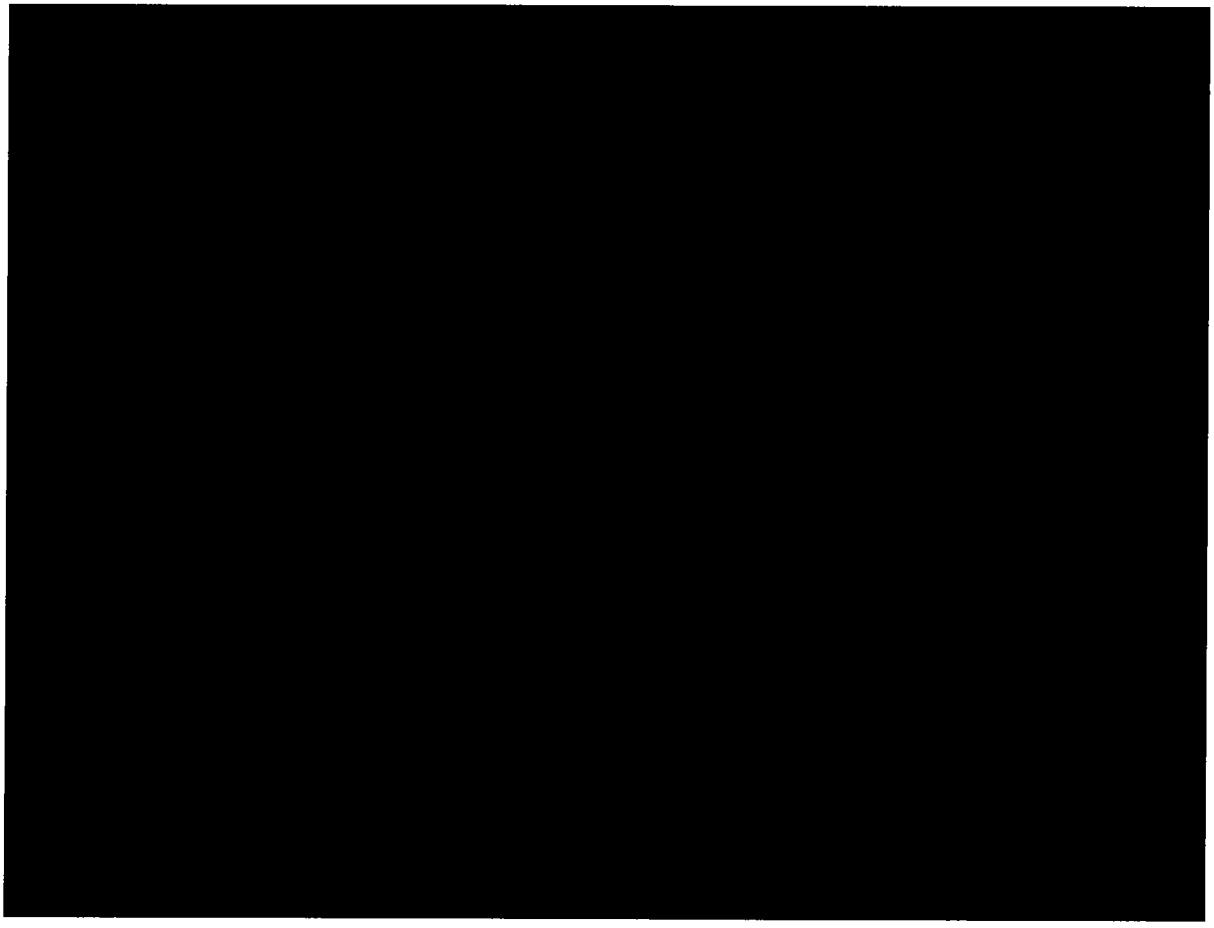
To view the main idea for this disclosure, click on this doclink ---> (If you are prompted to enter a server name, please enter D01DB068/01/A/IBM)

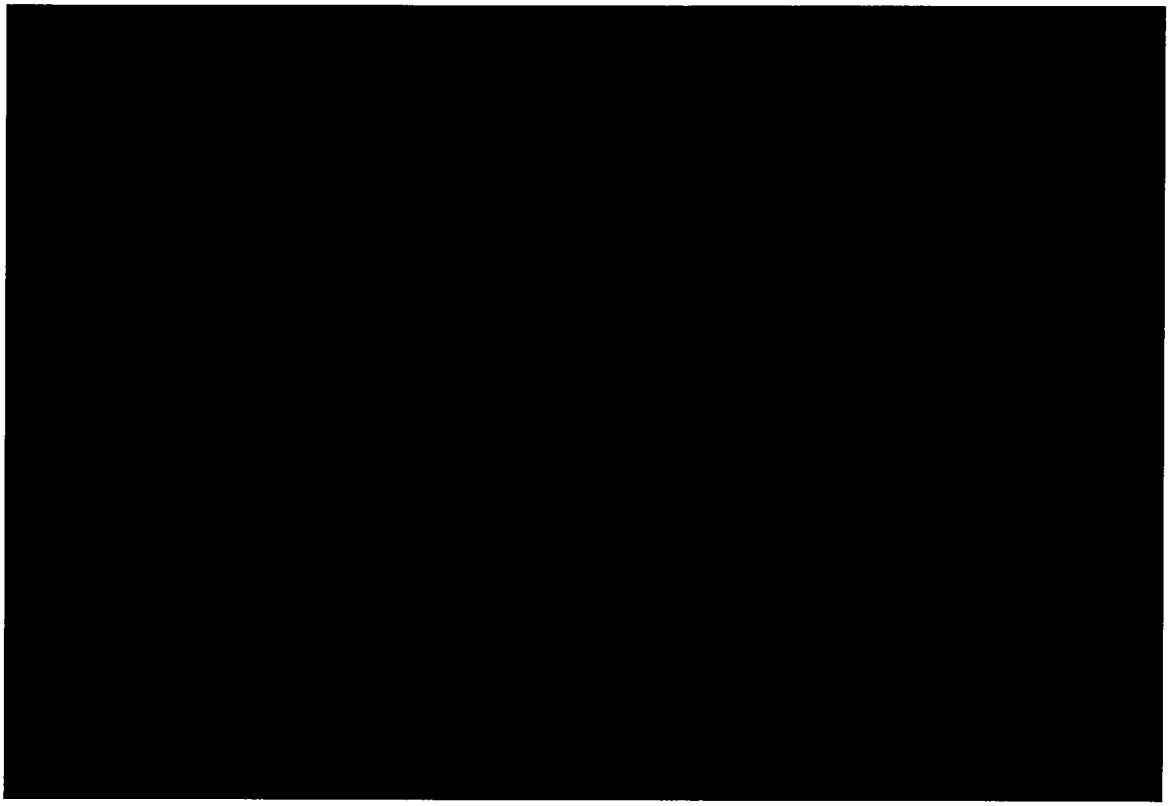
[redacted]

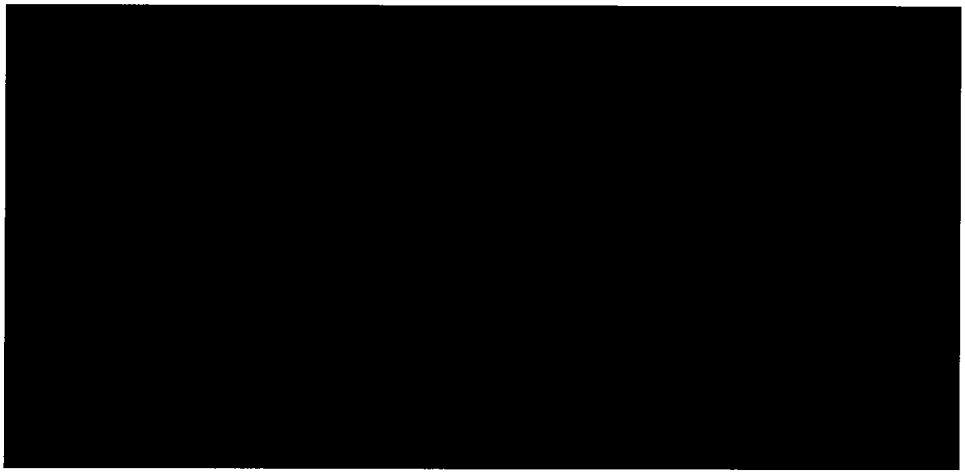
[redacted]

[redacted]









Evaluation
Search Information
Search Office Information
Final Decision
Post Disclosure Text & Drawings

Form Revised 09/01/02)

Main Idea for Disclosure YOR8-2003-0292

Prepared for and/or by an IBM Attorney - IBM Confidential

Archived On [redacted]

Title of disclosure (in English)
Virtualization of Network Resources for Distributed Interactive Applications

Main Idea of disclosure
1. Background: What is the problem solved by your invention? Describe known solutions to this problem (if any). What are the drawbacks of such known solutions, or why is an additional solution required? Cite any relevant technical documents or references.
Several large scale distributed interactive applications require support for SLA guarantees for better overall performance, and
often for the correctness of the applications. For example, in Grid/OGSA enviorment, it may be desirable to find servers that are
closest to the users in terms of round trip time. It may also be required to provide different server allocation strategies based on the user
class and application types. Currently most often distributed applications do not account for network conditions in allocating resources.
Also the network communication requirements are handled ground up by these applications even though there are lot of similarities in
the primitives provided by these bundled network layers.

In this invention, we propose middleware based network resource virtualization that manages the communication requirements accounting
for the resource conditions. Building upon the multi-type feature vector specifications, it proivdes an efficient method to handle heterogeneity
in SLA requirements among applications and users. It decouples applications from the actual location of the resources and network
conditions. The benefit of our approach is that the provided method is general enough to be used as a middleware for several distributed
interactive applications. Also it allows multi-type attributes to specify the SLA of a user that could combine the attributes from application
space, network metrics and user preferences etc.

Prior work in this area have focussed on very application specific approaches that are difficult to use in another application. Also these are
unable to leverage on various tradeoffs possible across different attribute types while accounting for SLAs. The work by [Bala2000],
[Padmanabhan2001] are steps in this direction that propose geographic based network level clustering to support communications.

2. Summary of Invention: Briefly describe the core idea of your invention (saving the details for questions #3 below). Describe the advantage(s) of using your invention instead of the known solutions described above.

The core idea is as follows: We assume that network metrics, user preferences, and application specific attributes are specified. We do not
assume any specific method for either collecting these or managing these attributes. However for completeness, we refer to our disclosure
on multi-type attribute feature vector for managing these attributes in a scalable manner. There are three

types of agents in this architecture
that we propose. The resources of the middleware that provides virtualization, application resources that server the application users and
users who actually use the resources. The middleware resources facilitate the communication that is required among the application servers
and application users. The middleware resources may be centralized or distributed. These resources may be managed using a hierarchical
control structure. Based on the attributes described earlier, the application servers are indexed in a manner to reflect the positioning in the
attribute space. Next, when an application user arrives, it is indexed accordingly to reflect its association with that part of the space that
matches its requirements. The application servers only see the indexed association as far as the real user is concerned. This association is
maintained at the middleware resource and not revealed to the application servers. If there is significant change in the set of attributes that
affect user's perceived SLA, then remappig of the user is performed. This separation from the actual users from the viewpoint of application
server enables the application as well as users to be allocated the appropriate resources. There are two issues need to be addressed in
our invention.

1. First, how to handle communication given that application servers do not have the actual identity of the users.
2. Second, how to handle failures of middleware resources as they are in the path between the application servers and the users.

Let us focus on the first aspect. The application servers shall use the indexed association to refer to an user. Based on the application
logic and session information, the application server would determine a set of indexed users that need to be communicated for any
event that has to be notified to the users. Given that set of users, a communication initiation is done via the middleware resource. The
middleware resource would then know how to best communicate to these set of users. Thus, two set of responsibilities have to handled
by the middleware resources. One is to establish the routing path for a given set of users. Second is to provide the APIs to the application
servers to specify the communication requirements. We shall describe the details of how these are supported in our invention in the
next section.

The second aspect is about making the method resilient to faults. This is managed by soft state based approach in wich periodic messages
are sent from the users and application servers to the middleware resources and via alternate path selection at the middleware servers. The
details are described in the later section.

3. Description: Describe how your invention works, and how it could be implemented, using text, diagrams and flow charts as appropriate.

The following are the details of each step that we described earlier.

1. Indexing of application servers and application users:

(i) Bootstrap phase: application server indexing

[redacted]

In this phase, a given set of application servers are indexed in a way to provide the concept of a virtual space. For example, an
Eucledean space may be used partition the application servers into a set of clusters and index each cluster using appropriate
number of bits. Next, within each cluster, an index is used to refer to each server. Note that the exact attribute that would be used to create
this indexing would possibly depend upon the application requirements etc. This clustering approach would also account for removal
and addition of application servers. There are several ways to achieve this in a clustering method.

(ii) Adaptation of bootstrap: This step involves reclustering in the advent of severe changes in the measured attributes that are used
for clustering in the step 1 (i). We assume that such a situation is a rare event and only happens at a very large time scale. If it is the
case that there is a large deviation in the measured attributes, the simple solution that we propose is to recompute the clustering and
reassing the indexing. Howerver, it does not preclude any incremental approaches that tries not to disturb the assigned indexing.

(iii) Application user indexing

When a user arrives into the system to use an application, its request is redirected to a middleware server. Based on its requirements and
its positioning in the application attribute space, a cluster is determined for the association with the virtual space. For example, this
association may reflect the nearest server to the user. Based on the indexing method, it is assigned an index that is exposed to the
application server. Another illustration could be AS-level clustering to figure out the index that should be assigned to the user. Given this
attribute aware indexing, an appropriate application server is then notified the presence of this user.

2. Management of association at the middleware servers

An entry is maintained in the middleware server to keep track of this association of index with actual IP address for every user. The actual
server that keeps this association information is determined according to the control hierarchy that manages the set of middleware servers.
In addition, middleware server control protocol may decide to replicate this information at a set of other middleware servers. The user is also
returned a descriptor that reflects the middleware server it should contact for further communication with the application space.

3. Routing path construction

The application logic and the session management module together may determine a set of users that have to be notified for any event.
The application uses the APIs to describe this either to the middleware server or through session management module. Given the notification
event and a set of users, middleware server determines the actual identity of the users that have to contacted. It may use shared overlay routing
infrastructure or per-session overlay tree for communication. It shall support both types of routing.

Shared Routing Infrastructure: Each middleware server maintains routing entries that determine which

[redacted]

server to send messages for which
prefix. One possible approach is to use a hierarchical tree to manage this routing. Other approach would be to use a set of finger pointers
to manage the routing among the set of middleware servers. If the message is destined for an user in the same cluster as itself, it
forwards this message directly. If not, it uses the shared routing infrastructure to forward this message to appropriate server for further
action.

Per-session overlay infrastructure: Given a set of users, a separate overlay tree is established to route messages along this tree. It does
not require any specific algorithm to construct this tree - however as preferred embodiment, we propose hierarchical tree based routing.
The details of this construction is as follows: all the middleware nodes that have any users in their domain would participate as parent
nodes in this tree. They form an overlay path among themselves. Now within a domain, another tree is constructed among the users with
the tree routed at the middleware server node that is assigned for this domain. Note that while constructing this per-session tree, the
specified SLAs are taken into account to construct the overlay tree.

4. Handling middleware server failures

Given that association of users and application servers are maintained at the middleware servers, it is crucial to handle middleware
node failures. This is achieved by two set of mechanisms in our architecture. The first one is to replicate the states at multiple middleware
nodes besides the designated server. Second, each user uses keep alive messages to check for node failures. In the advent of
node failures, alternate servers are contacted to re-establish the connection. While the above approach is our preferred
embodiment, it is not required to use this method to support resiliency.